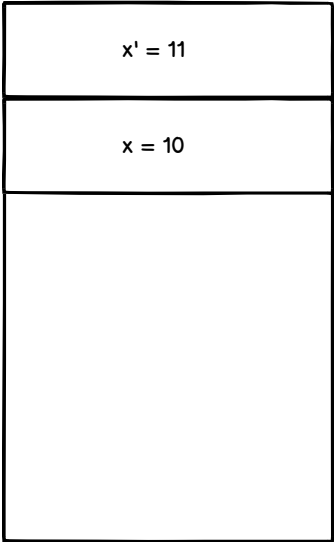


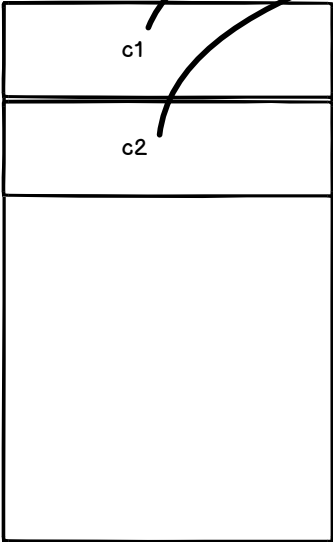
Typy referencyjne a wartosciowe

```
public static void Main();
{
    int x = 10;
    Add(x);
    Console.WriteLine(x);
}
```

```
public void Add(int x)
{
    x++;
    Console.WriteLine(x);
}
```



STOS



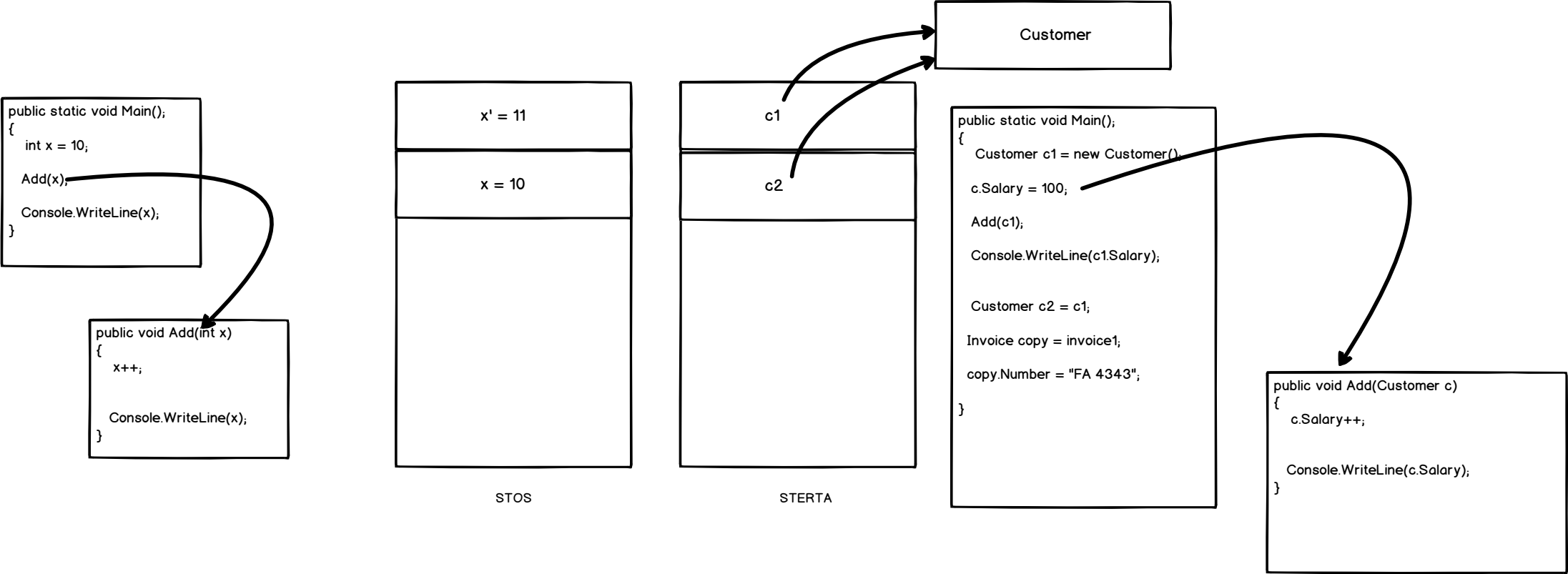
STERTA



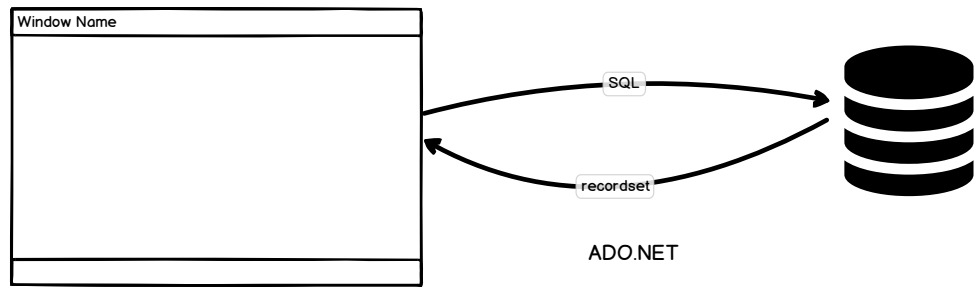
```
public static void Main();
{
    Customer c1 = new Customer();
    c.Salary = 100;
    Add(c1);
    Console.WriteLine(c1.Salary);

    Customer c2 = c1;
    Invoice copy = invoice1;
    copy.Number = "FA 4343";
}
```

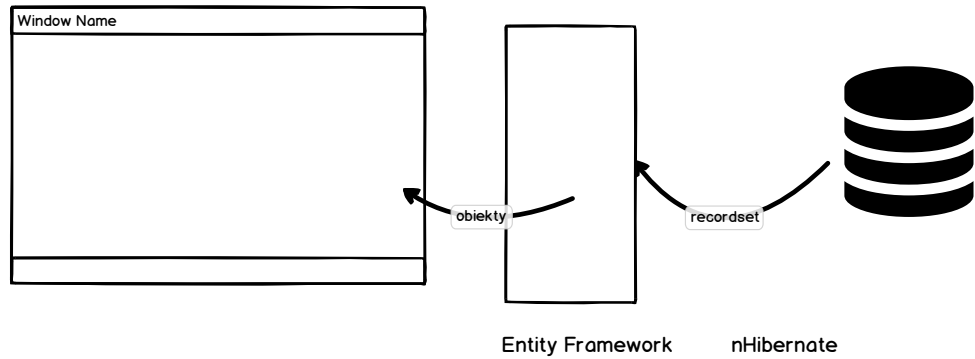
```
public void Add(Customer c)
{
    c.Salary++;
    Console.WriteLine(c.Salary);
}
```



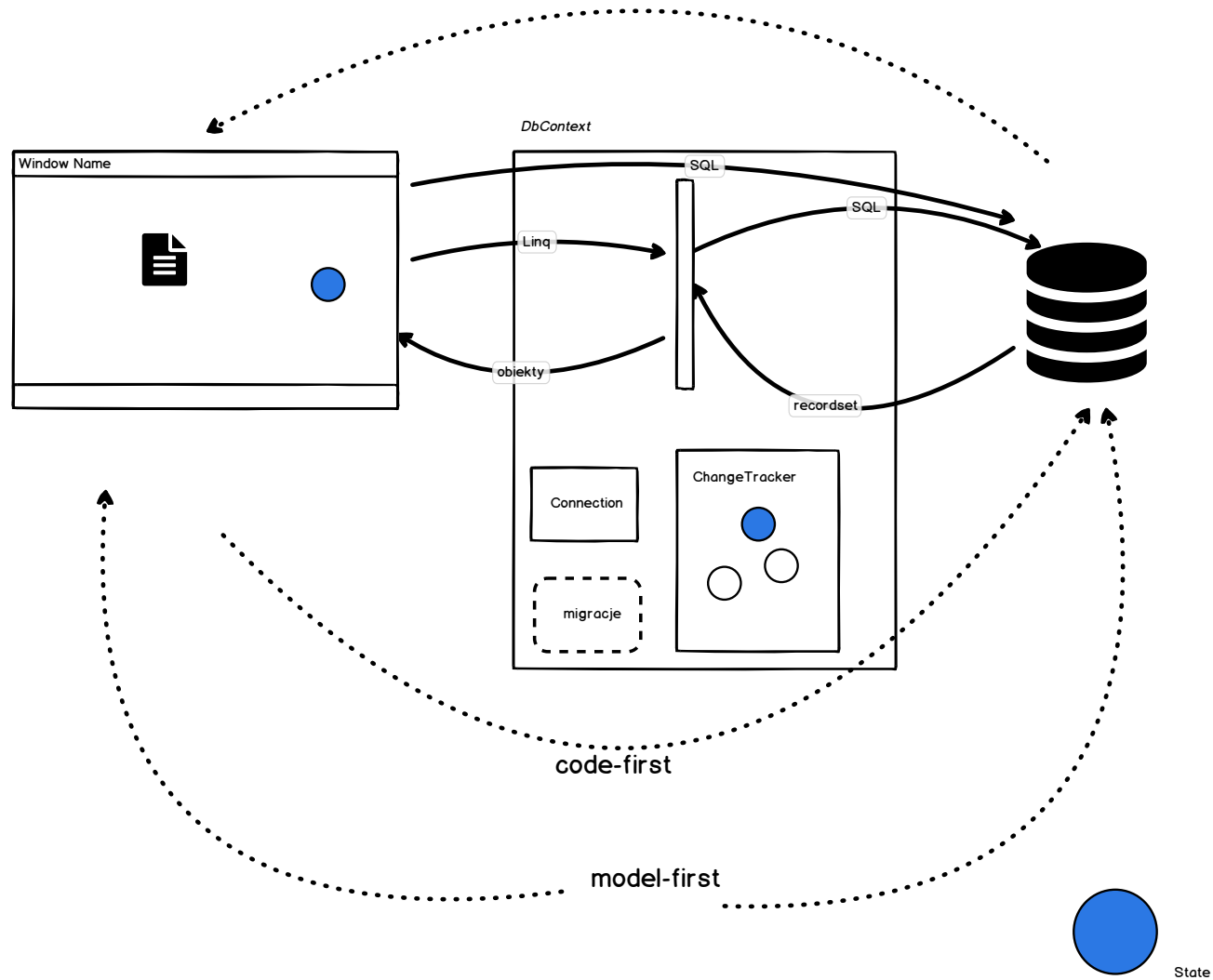
ADO.NET



ORM (Object Relation Mapping)



database-first



Window Name

C#

LINQ

Extension methods

delegates -> Wyrażenia lambda

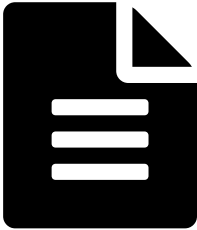
ADO.NET

SQL

C#

XPath

System.OI



Typy referencyjne a wartosciowe copy

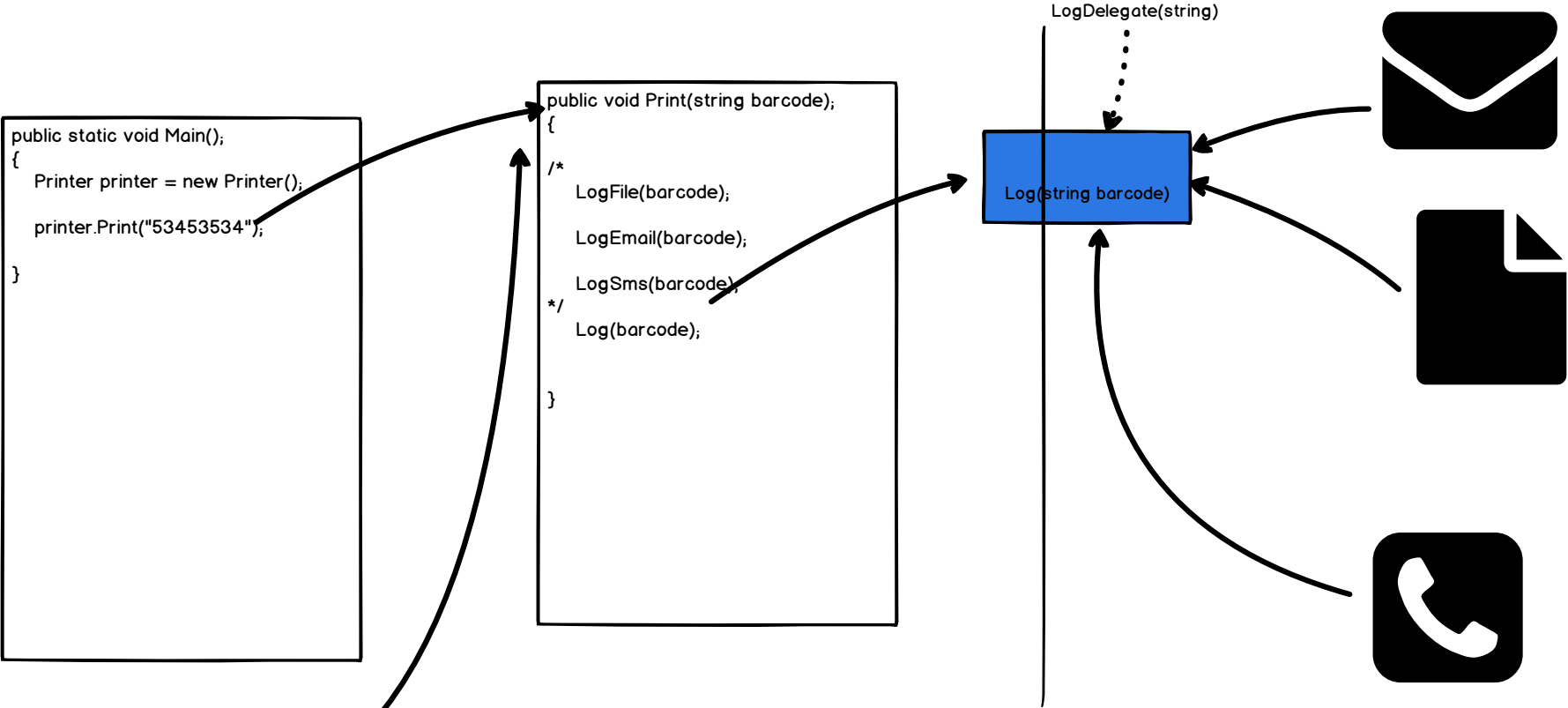
```
public static void Main();
{
    Printer printer = new Printer();
    printer.Print("53453534");
}
```

```
public static void Main();
{
    Printer printer = new Printer();
    printer.Print("4664564");
}
```

```
public void Print(string barcode);
{
    /*
        LogFile(barcode);
        LogEmail(barcode);
        LogSms(barcode);
    */
    Log(barcode);
}
```

Log(string barcode)

LogDelegate(string)



## Lambda

```
public int Add(int x, int y)
{
    return x + y;
}
```

wyrażenia lambda

```
int x = 10;
int y = 5;
```

```
(x, y) => x + y
```

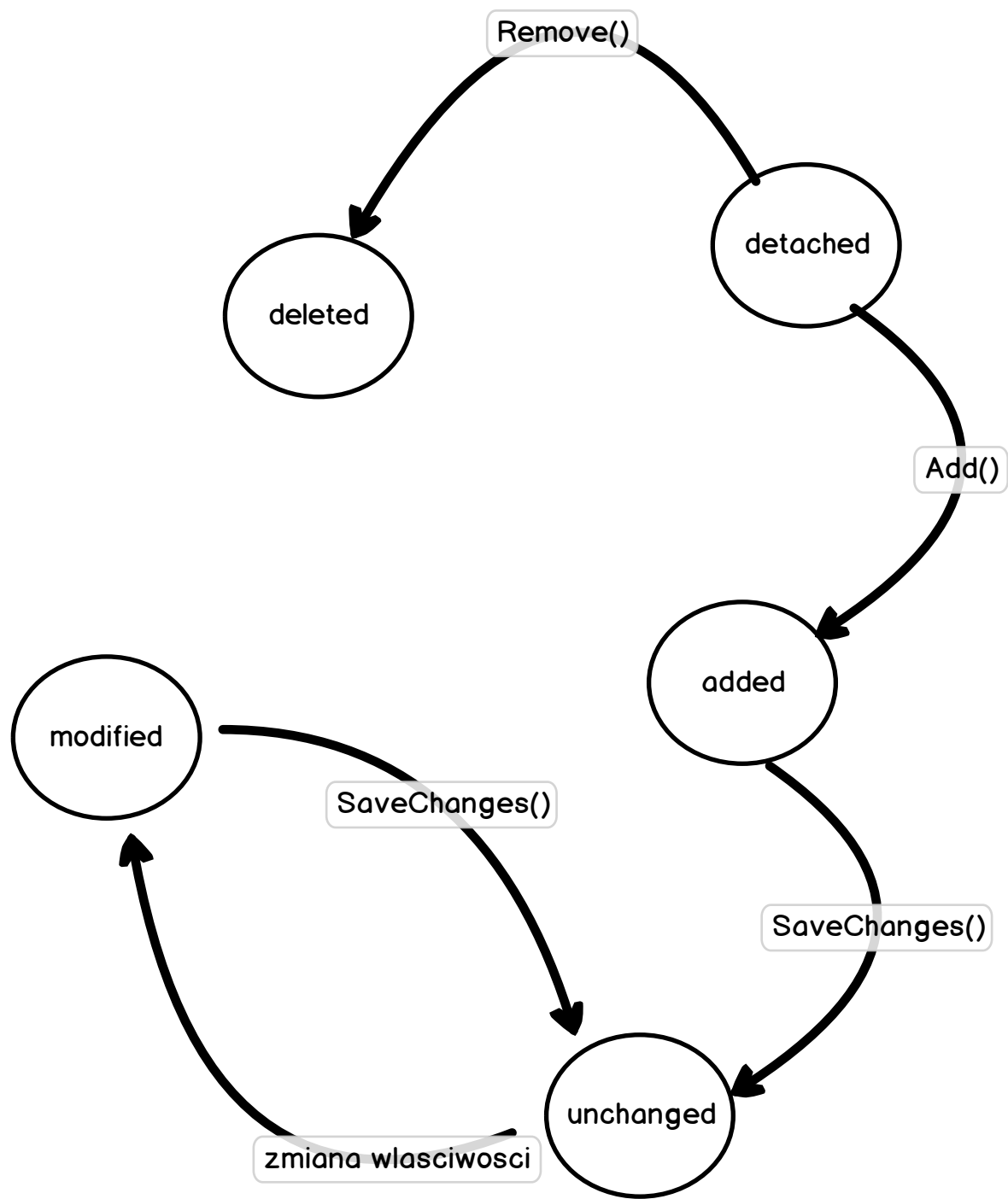
```
public int ____ (int x, int y)
{
    return x + y;
}
```

$$f(x, y) = x + y \quad x, y \in \mathbb{C}$$

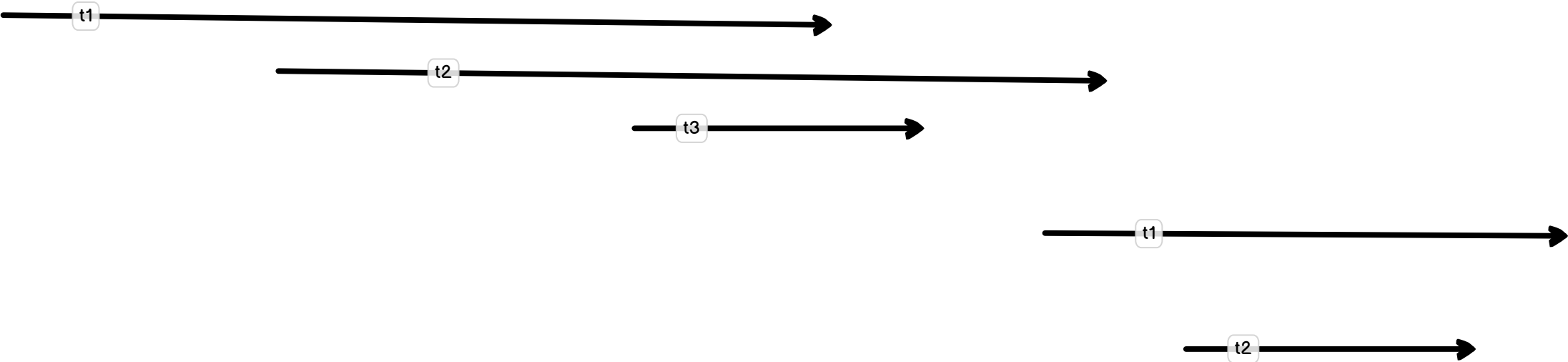
rachunek lambda

$$(x, y) \rightarrow x + y \quad x, y \in \mathbb{C}$$

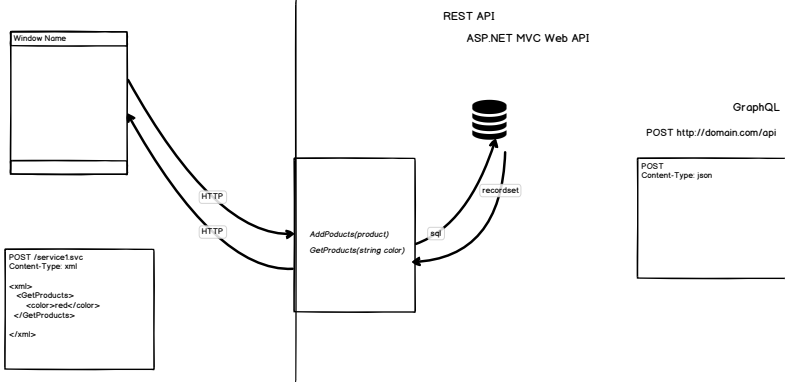
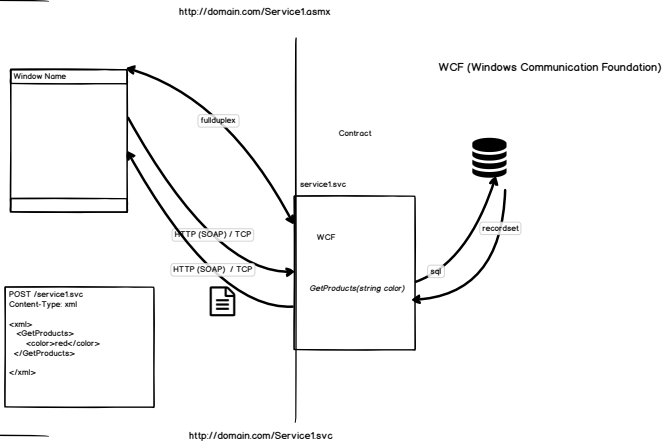
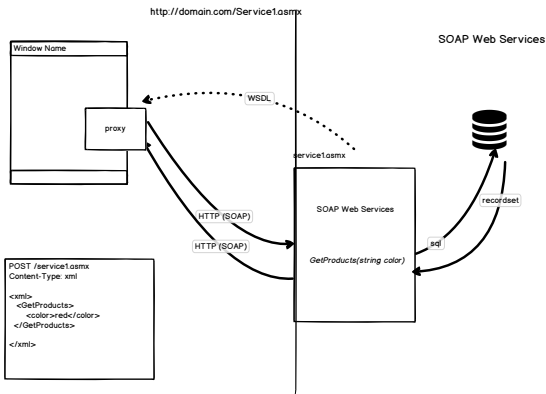
Entity States



Threads



## Threads copy



GET	pobierz	GET http://domain.com/api/products
POST	utworz	GET http://domain.com/api/products/10
PUT	zamien	GET http://domain.com/api/products/red
DELETE	usun	GET http://domain.com/api/products?from= <u>10</u> to=400;color=red
PATCH	zmien	GET http://domain.com/api/products/10/customers

DELETE <http://domain.com/api/products/10>

```
POST http://domain.com/api/products
Content-Type: xml

<Product>
  <name>Produkt 1</name>
  <color>red</color>
</Product>
```

```
POST http://domain.com/api/products
Content-Type: json

{"name": "Produkt 1", "color": "red"}
```

